Logspace computability and regressive machines

Stefano Mazzanti

Dipartimento di Culture del Progetto
Università Iuav di Venezia
Fondamenta delle Terese 2206, 30123 Venezia, Italy
email: mazzanti@iuav.it

Abstract. We consider the function class **E** generated by the constant functions, the projection functions, the predecessor function, the substitution operator, and the recursion on notation operator. Furthermore, we introduce regressive machines, i.e. register machines which have the division by 2 and the predecessor as basic operations. We show that **E** is the class of functions computable by regressive machines and that the sharply bounded functions of **E** coincide with the sharply bounded logspace computable functions.

Keywords: recursion on notation, logspace computable functions.

1 Introduction

One of the main features of logspace-algorithms is their incapacity to copy the whole input due to memory shortage, thus forcing them to read the data "on the fly", every time they are needed.

This feature has been exploited in [3] where the set **L** of logspace computable predicates has been shown to be the set of predicates recognized by read-only while programs and in [4] where the closure with respect to substitution and simultaneous recursion on notation of the constant functions and the projection functions has been shown to contain the characteristic functions of the predicates in **L**.

In this paper, we consider the class **E** of number theoretic functions defined as the closure with respect to substitution and (unbounded) recursion on notation of the predecessor function, the constant functions and the projection functions.

We show that **E** is a subset of the logspace computable functions which contains all the sharply bounded logspace functions. Moreover, we show that **E** is the set of functions computable by regressive machines, a kind of register machines which have the division by 2 and the predecessor as basic operations on registers.

Therefore, the present work can be considered an improvement of the characterization of **L** given in [4] because we use recursion on notation instead of simultaneous recursion and we characterize not only the $\{0,1\}$-valued logspace functions, but also the sharply bounded logspace computable functions.

2 Preliminaries

In this paper, we will only consider functions with finite arity on the set $\mathbb{N} = \{0, 1, \ldots\}$ of natural numbers.

From now on, we agree that x, y, z, i, j, n range over $\mathbb{N}$, that a, b, c, k range over $\mathbb{N} - \{0\}$, that $\mathbf{x}, \mathbf{y}, \mathbf{z}$ range over sequences (of fixed length) of natural numbers, that p, q range over integer polynomials with nonnegative coefficients and that f, g, h range over functions.

A function f is a *polynomial growth function* iff there is a polynomial p such that $|f(\mathbf{x})| \leq p(|\mathbf{x}|)$ for any $\mathbf{x}$, where $|x_1, \ldots, x_n| = |x_1|, \ldots, |x_n|$ and $|x| = \lceil \log_2(x+1) \rceil$ is the number of bits of the binary representation of x.[1] Moreover, f is *sharply bounded* iff there is a polynomial p such that $f(\mathbf{x}) \leq p(|\mathbf{x}|)$ for any $\mathbf{x}$, and f is *regressive* iff there is some constant k such that $f(\mathbf{x}) \leq \max(\mathbf{x}, k)$ for any $\mathbf{x}$, see [2].

We will use the following unary functions: the *binary successor* functions $s_0 : x \mapsto 2x$ and $s_1 : x \mapsto 2x + 1$; the *constant* functions $C_n : x \mapsto n$ for any $n \in \mathbb{N}$; the *division by two* function $div_2 : x \mapsto \lfloor x/2 \rfloor$; the *remainder* function $rem_2 : x \mapsto x - 2\lfloor x/2 \rfloor$; the *length* function $len : x \mapsto |x|$.

We will also use the following functions: the *modified subtraction* function $sub : x, y \mapsto x \dot{-} y = \max(x - y, 0)$; the predecessor function $P : x \mapsto x \dot{-} 1 = \max(x - 1, 0)$; the *bit* function $bit : x, y \mapsto rem_2(\lfloor x/2^y \rfloor)$; the *smash* function $smash : x, y \mapsto x \# y = 2^{|x| \cdot |y|}$; the *most significant part* function $MSP : x, y \mapsto \lfloor x/2^y \rfloor$. Recall that $bit(x, y)$ is the y-th bit of x and that $MSP(x, y)$ is the number represented in binary by the $|x| \dot{-} y$ leftmost bits of x. Finally, we will use the *substitution* operator $SUBST(g_1, \ldots, g_b, h)$ transforming functions $g_1, \ldots, g_b : \mathbb{N}^a \to \mathbb{N}$ and function $h : \mathbb{N}^b \to \mathbb{N}$ into the function $f : \mathbb{N}^a \to \mathbb{N}$ such that $f(\mathbf{x}) = h(g_1(\mathbf{x}), \ldots, g_b(\mathbf{x}))$ and the *recursion on notation* operator $RN(g, h_0, h_1)$ transforming function $g : \mathbb{N}^a \to \mathbb{N}$ and functions $h_0 : \mathbb{N}^{a+2} \to \mathbb{N}$ and $h_1 : \mathbb{N}^{a+2} \to \mathbb{N}$ into the function $f : \mathbb{N}^{a+1} \to \mathbb{N}$ such that

$$f(0, \mathbf{y}) = g(\mathbf{y}),$$
$$f(s_i(x), \mathbf{y}) = h_i(x, \mathbf{y}, f(x, \mathbf{y}))$$

where $i \in \{0, 1\}$. Let $\mathbf{E}$ be the closure under substitution and recursion on notation of the set comprehending the predecessor function P, the constant functions C_n for any n and the projection functions $I^a[i] : x_1, \ldots, x_a \mapsto x_i$ $(1 \leq i \leq a)$ with any arity a.

3 Regressive machines

Now, we introduce *regressive machines*, a kind of random access machines which have the division by 2 and the predecessor as basic operations.

[1] Here, the notation $|x_1, \ldots, x_n|$ is just an abbreviation for the sequence $|x_1|, \ldots, |x_n|$. However, $|x_1, \ldots, x_n|$ is also interpreted as $|x_1| + \ldots + |x_n|$.

A regressive machine operates on a finite number of variables (the registers) $\mathtt{X_1}, \ldots, \mathtt{X_b}$ and its program is built up according to the following rules:[2]

$$\mathtt{P} ::= \mathtt{X_i} := \mathtt{e} | \mathtt{pred}(\mathtt{X_i}) | \mathtt{half}(\mathtt{X_i}) | \mathtt{P_1}; \mathtt{P_2} | \mathtt{loop}\, \mathtt{X_i}\, \mathtt{do}\, \mathtt{P}\, \mathtt{end}$$

where instruction $\mathtt{X_i} := \mathtt{e}$ sets the value of register $\mathtt{X_i}$ to the value of expression $\mathtt{e}$, and $\mathtt{e}$ can be any natural number constant, any register, or the least significant bit $\mathtt{lsb}(\mathtt{X_j})$ of register $\mathtt{X_j}$. Moreover, instructions $\mathtt{pred}(\mathtt{X_i})$ and $\mathtt{half}(\mathtt{X_i})$ compute the predecessor and (the quotient of) the division by 2 of $\mathtt{X_i}$, respectively. Finally, the program $\mathtt{loop}\, \mathtt{X_i}\, \mathtt{do}\, \mathtt{P}\, \mathtt{end}$ executes $|x|$ times program $\mathtt{P}$, where x is the value held by $\mathtt{X_i}$.

From now on, we adopt the notation of [4], so that registers and programs are denoted by capital letters and we will write programs with the typewriter font.

For any program $\mathtt{P}$ with b registers, a memory state of $\mathtt{P}$ is a sequence $\mathbf{x} = x_1, \ldots, x_b$ where x_i is the value of $\mathtt{X_i}$.

Consider the function $m_{\mathtt{P}} : \mathbb{N}^b \to \mathbb{N}^b$ such that $m_{\mathtt{P}}(\mathbf{x})$ is the state of $\mathtt{P}$ after the computation of $\mathtt{P}$ starting from state $\mathbf{x}$. The following lemma states that regressive machines compute regressive functions and that they operate in polynomial time as long as instructions are executed sequentially and each operation is executed in constant time.

Lemma 1. *For any program* $\mathtt{P}$ *with* b *registers there is a constant* c *such that* $I^b[i](m_{\mathtt{P}}(\mathbf{x})) \leq \max(\mathbf{x}, c)$ *for any* $1 \leq i \leq b$ *and there is a polynomial* p *such that the running time of* $\mathtt{P}$ *starting from memory state* $\mathbf{x}$ *is bounded by* $p(|\mathbf{x}|)$.

A program $\mathtt{P}$ with b registers computes a function $f : \mathbb{N}^a \to \mathbb{N}$ with respect to input registers $\mathtt{X_1}, \ldots, \mathtt{X_a}$ and output register $\mathtt{X_j}$ iff for any $x_1, \ldots, x_a$ the value $f(x_1, \ldots, x_a)$ is returned in register $\mathtt{X_j}$ when $\mathtt{P}$ is executed with $\mathtt{X_i}$ having initial value x_i for $1 \leq i \leq a$ and all the other variables are initialized to zero.

4 Main results

Let **SB** be the set of sharply bounded functions, let **RM** be the set of functions computable by regressive machines and let **FL** be the set of logspace computable functions. The following statement summarizes the relationships between logspace functions, class **E** and regressive machines.

Theorem 1.

$$\mathbf{FL} \cap \mathbf{SB} \subseteq \mathbf{E} \subseteq \mathbf{RM} \subseteq \mathbf{FL} \cap \mathbf{E}.$$

[2] The loop predecessor machines of [2] do not have either the assignment `X:=lsb(Y)` or the division instruction, and use the `LOOP X DO P` construct, which iterates x many times `P`, where x is the value held by `X`. In [3], counter machines with only decrement instructions have been considered, but the registers' contents are bounded by the length of the input.

Proof (Sketch). We first show that class **E** contains sharply bounded versions of the arithmetic operations (e.g. $add_p(x,y,z) = y+z$ for $y,z \leq p(|x|)$) and is closed with respect to the *sharply bounded maximization* operator $MAX_p(g)$ transforming function $g : \mathbb{N}^{a+1} \to \mathbb{N}$ into the function $f : \mathbb{N}^a \to \mathbb{N}$ such that $f(\mathbf{x}) = \max\{i \leq p(|\mathbf{x}|) | g(\mathbf{x},i) \neq 0\}$ if $\{i \leq p(|\mathbf{x}|) | g(\mathbf{x},i) \neq 0\} \neq \emptyset$, otherwise $f(\mathbf{x}) = 0$. Then, we set $bit_f(\mathbf{x},i) = bit(f(\mathbf{x}),i)$ and show that $bit_f \in \mathbf{E}$ for any $f \in \mathbf{FL}$. The proof is carried out by induction on the characterization of **FL** given by Clote and Takeuti [1] which defines **FL** as the least function class closed with respect to substitution, concatenation recursion on notation and sharply bounded recursion on notation of the set comprehending the projection functions, the binary successor functions, the bit, the length and the smash functions.

Then, the first inclusion of the theorem is true because for any $g \in \mathbf{FL}$ such that $g(\mathbf{x}) \leq p(|\mathbf{x}|)$ for some polynomial p, we have

$$g(\mathbf{x}) = \max\{y \leq p(|\mathbf{x}|) | \forall_{i<|p(|\mathbf{x}|)|} bit(y,i) = bit_g(\mathbf{x},i)\}$$

and the characteristic function of the predicate $\forall_{i<|p(|\mathbf{x}|)|} bit(y,i) = bit_g(\mathbf{x},i)$ is in **E**. The second inclusion can be easily obtained by showing (by induction on **E**) that for any function $f \in \mathbf{E}$ there is a regressive machine computing f.

To show the third inclusion, we introduce counter machines and show that they simulate regressive machines using only a logarithmic amount of memory space. Then, counter machines can be easily simulated by functions in **FL**∩**E** and we obtain that $\mathbf{RM} \subseteq \mathbf{FL} \cap \mathbf{E}$. A *counter machine* operates on a finite number of read-only input registers and a finite number of read/write registers called *counters*. Input registers are denoted as $\mathtt{Y_1}, \ldots, \mathtt{Y_a}$ and counters are denoted as $\mathtt{Z_1}, \ldots, \mathtt{Z_b}$ for some a and b. Let $\mathbf{y} = y_1, \ldots, y_a$ be the input values and let $\mathbf{z} = z_1, \ldots, z_b$ be the values of the counters. A counter machine program is defined according to the following rules:

$$\mathtt{Q} ::= \mathtt{Z_i} := \mathtt{e} | \mathtt{succ(Z_i)} | \mathtt{half(Z_i)} | \mathtt{Q_1; Q_2} | \mathtt{if\ (e_1 = n)\ then\ Q_1\ else\ Q_2}$$
$$| \mathtt{loop\ E_i\ do\ Q_1\ end}$$

where $\mathtt{e}$ is any constant, any counter or $\mathtt{lsb(E_j)}$, expression $\mathtt{e_1}$ can be $\mathtt{Z_i}$, $\mathtt{bit(Y_{Z_i}, Z_j)}$ or $\mathtt{lsb(Z_i)}$, and $\mathtt{E_i}$ is an expression whose value is

$$e_i(\mathbf{y},\mathbf{z}) = \begin{cases} z_{i+2} & \text{if } z_i = 0, \\ MSP(y_{z_i}, z_{i+1}) \dot{-} z_{i+2} & \text{otherwise} \end{cases} \quad (1 \leq i \leq b-2).$$

Furthermore, we define the function $M_{\mathtt{Q}} : \mathbb{N}^{a+b} \to \mathbb{N}^a$ such that $(\mathbf{y}, M_{\mathtt{Q}}(\mathbf{y},\mathbf{z}))$ is the memory state returned after the computation of a counter machine program $\mathtt{Q}$ starting from the state $(\mathbf{y},\mathbf{z})$. Then, every regressive machine program $\mathtt{P}$ with b registers is *simulated* by a counter machine program $\mathtt{Q}$ with $3b$ registers. This means that for any $\mathbf{x} \in \mathbb{N}^b$, $\mathbf{y} \in \mathbb{N}^a$ and $\mathbf{z} \in \mathbb{N}^{3b}$, if $I^b[i](\mathbf{x}) = e_{3i-2}(\mathbf{y},\mathbf{z})$ for any $1 \leq i \leq b$, then $I^b[i](m_{\mathtt{P}}(\mathbf{x})) = e_{3i-2}(\mathbf{y}, M_{\mathtt{Q}}(\mathbf{y},\mathbf{z}))$ for any $1 \leq i \leq b$. In other words, the value of register $\mathtt{X_i}$ of program $\mathtt{P}$ is represented by counters $\mathtt{Z_{3i-2}}, \mathtt{Z_{3i-1}}, \mathtt{Z_{3i}}$ of program $\mathtt{Q}$ so that $e_{3i-2}(\mathbf{y},\mathbf{z}) = x_i$. If $\mathtt{X_i}$ has been set

to a constant value, then $z_{3i-2} = 0$ and z_{3i} is the value of $\mathtt{X_i}$. Otherwise, an input value has been assigned (or copied) to $\mathtt{X_i}$ and decrement or division instructions have been performed on it. In that case, the value of $\mathtt{X_i}$ is $MSP(y_{z_{3i-2}}, z_{3i-1}) \dot{-} z_{3i}$ (z_{3i-2} is the index of the input value, z_{3i-1} is the number of divisions and z_{3i} is less than or equal to the number of decrements). Then, $\mathtt{pred(X_i)}$ is simulated by $\mathtt{succ(Z_{3i})}$ (if $\mathtt{X_i}$ is positive) whereas $\mathtt{half(X_i)}$ is simulated by $\mathtt{succ(Z_{3i-1}); half(Z_{3i})}$ ($\mathtt{Z_{3i}}$ could also be increased according to its parity and the value of $bit(y_{z_{3i-2}}, z_{3i-1})$).

So, for any function $f : \mathbb{N}^a \to \mathbb{N}$ computed by a regressive machine program $\mathtt{P}$ with b registers, there is a counter machine $\mathtt{Q}$ with $3b$ counters such that

$$f(\mathbf{x}) = e_{3j-2}(\mathbf{x}, M_{\mathtt{Q}}(\mathbf{x}, 1, 0, 0, \ldots, a, 0, 0, \ldots, 0))$$

where j is the index of the output register of $\mathtt{P}$. Moreover, by Lemma 1 there is a polynomial p such that $p(|\mathbf{x}|)$ bounds all the counters at any step of the computation of $\mathtt{Q}$. Therefore, we encode the counters with a single number $c_p(\mathbf{x}, \mathbf{z}) = z_1 p(|\mathbf{x}|)^{(3b-1)} + \ldots + z_{3b-1} p(|\mathbf{x}|) + z_{3b} < p(|\mathbf{x}|)^{3b}$ and we define functions $\tilde{e}_{p,i}, \tilde{M}_{p,\mathtt{Q}} : \mathbb{N}^{a+1} \to \mathbb{N}$ belonging to $\mathbf{FL} \cap \mathbf{E}$ such that $\tilde{e}_{p,i}(\mathbf{x}, c_p(\mathbf{x}, \mathbf{z})) = e_i(\mathbf{x}, \mathbf{z})$, $\tilde{M}_{p,\mathtt{Q}}(\mathbf{x}, c_p(\mathbf{x}, \mathbf{z})) = c_p(M_{\mathtt{Q}}(\mathbf{x}, \mathbf{z}))$ and

$$f(\mathbf{x}) = \tilde{e}_{p,3j-2}(\mathbf{x}, \tilde{M}_{p,\mathtt{Q}}(\mathbf{x}, c_p(\mathbf{x}, 1, 0, 0, \ldots, a, 0, 0, \ldots, 0))).$$

Since $c_p \in \mathbf{FL} \cap \mathbf{E}$, we obtain that $f \in \mathbf{FL} \cap \mathbf{E}$.

From Theorem 1 we obtain immediately that $\mathbf{E}$ is a subset of logspace computable functions and coincides with the class of functions computable by regressive machines.

Corollary 1. $\mathbf{E} = \mathbf{RM} \subseteq \mathbf{FL}$.

Moreover, by Theorem 1, we are also able to state that the sharply bounded logspace functions coincide with the sharply bounded functions in $\mathbf{E}$.

Corollary 2. $\mathbf{FL} \cap \mathbf{SB} = \mathbf{E} \cap \mathbf{SB}$.

Finally, from the corollary above we obtain the following new characterization of $\mathbf{L}$.

Corollary 3. *The characteristic functions of logspace predicates coincide with the* $\{0,1\}$*-valued functions in* $\mathbf{E}$.

References

1. P. Clote and G. Takeuti, *First order bounded arithmetic and small boolean circuit complexity classes*, in *Feasible Mathematics II*, Birkhäuser Boston, 1995.
2. P. C. Fischer, J. C. Warkentin, *Predecessor Machines*, J. Comput. System Sci. 8 (1974) 190-219.
3. N. D. Jones, *LOGSPACE and PTIME characterized as programming languages*, Theoret. Comput. Sci. 228 (1999) 151-174.
4. L. Kristiansen, *Neat algebraic characterizations of LOGSPACE and LINSPACE*, Comput. Complexity 14 (2005) 72–88.